Computers & Operations Research 167 (2024) 106665

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

A stabilised Benders decomposition with adaptive oracles for large-scale stochastic programming with short-term and long-term uncertainty

Hongyu Zhang [a,*], Nicolò Mazzi [c], Ken McKinnon [b], Rodrigo Garcia Nava [b], Asgeir Tomasgard [a]

[a] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Høgskoleringen 1, 7491, Trondheim, Norway
[b] School of Mathematics, University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Road, Edinburgh, EH9 3FD, United Kingdom
[c] aHead-Research, Corso Svizzera 185, Torino, 10149, Italy

ARTICLE INFO

Keywords:
Large scale optimisation
Multi-stage stochastic programming
Multi-horizon stochastic programming
Stabilised Benders decomposition with adaptive oracles
Level method

ABSTRACT

Benders decomposition with adaptive oracles was proposed to solve large-scale optimisation problems with a column-bounded block-diagonal structure, where subproblems differ only in the right-hand side and cost coefficients. Adaptive Benders reduces computational effort significantly by iteratively building inexact cutting planes and valid upper and lower bounds. However, Adaptive Benders and standard Benders may suffer severe oscillation when solving degenerate models. Therefore, we propose stabilising Adaptive Benders with the level method and adaptively selecting which subproblems to solve each iteration for more accurate information. In addition, we propose a dynamic level method to improve the robustness of stabilised Adaptive Benders by adjusting the level set each iteration. We compare stabilised Adaptive Benders with the unstabilised versions of Adaptive Benders with one subproblem solved per iteration and standard Benders on a multi-region long-term power system investment planning problem with short-term and long-term uncertainty. The problem is formulated as multi-horizon stochastic programming. Four algorithms were implemented to solve linear programming with up to 1 billion variables and 4.5 billion constraints. The computational results show that: (a) for a 1.00% convergence tolerance, the proposed stabilised method is up to 113.7 times faster than standard Benders and 2.1 times faster than unstabilised Adaptive Benders; (b) for a 0.10% convergence tolerance, the proposed stabilised method is up to 45.5 times faster than standard Benders and unstabilised Adaptive Benders cannot solve the largest instance to convergence tolerance due to severe oscillation and (c) dynamic level method makes stabilisation more robust.

1. Introduction

In this paper, we propose an algorithm to efficiently solve a class of large-scale linear programming problems and address the degeneracy issue. We consider problems that exhibit a column-bounded block-diagonal structure, and in which Subproblems (**SP**s) differ only in the right-hand side and cost coefficients. Such problems can be formulated as a full Master Problem (**MP**),

$$\mathbf{MP}: \quad \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) + \sum_{i\in\mathcal{I}} \pi_i g(\mathbf{x}_i, \mathbf{c}_i), \tag{1}$$

where $f(\mathbf{x}) = \sum_{i\in\mathcal{I}} \pi_i \mathbf{c}_i^\top \mathbf{x}_i$ and the function $g(\mathbf{x}_i, \mathbf{c}_i)$ is the optimal solution of the linear programming subproblem,

$$\mathbf{SP}_i: \quad g(\mathbf{x}_i, \mathbf{c}_i) := \min_{\mathbf{y}_i\in\mathcal{Y}} \{ \left(\mathbf{c}_i^\top C + \mathbf{c}^\top\right) \mathbf{y}_i \,|\, A\mathbf{y}_i \le \mathbf{b} + B\mathbf{x}_i \}. \tag{2}$$

Here, the set of decision nodes is given by $\mathcal{I}$, the $\mathbf{x}_i$ are subvectors of $\mathbf{x}$, the $\mathbf{y}_i$ are the decision variables of $\mathbf{SP}_i$, which must lie in the polyhedral set $\mathcal{Y}$, the π_i are non-negative constants, the coefficient matrices A, B, and C are the same in all **SP**s. The parameters $\mathbf{c}_i^\top C + \mathbf{c}^\top$ and $\mathbf{b} + B\mathbf{x}_i$ are the cost coefficients and right hand side that are constant in the subproblem. The decisions, $\mathbf{x}$, made in the master problem are passed to **SP**s as right-hand side parameters. This formulation includes some forms of multi-stage stochastic programming problems, but the algorithm developed in this paper can be applied to any optimisation problems with the same structure.

Solving this problem directly can be computationally expensive. However, when $g(\mathbf{x}_i, \mathbf{c}_i)$ is decreasing w.r.t. $\mathbf{x}_i$, and increasing w.r.t. $\mathbf{c}_i$, one can exploit these properties to efficiently solve the problem (Mazzi et al., 2020). The monotonicity property can be natural of the problem, for example, when B and C are nonnegative matrices. If the original problem does not have the monotonicity property, it can be reformulated to have this property as shown by Mazzi et al. (2020). Mazzi

* Corresponding author.
E-mail addresses: hongyu.zhang@ntnu.no (H. Zhang), mazzi.nicolo@gmail.com (N. Mazzi), K.McKinnon@ed.ac.uk (K. McKinnon), Rodrigo.Garciana@ed.ac.uk (R.G. Nava), asgeir.tomasgard@ntnu.no (A. Tomasgard).

https://doi.org/10.1016/j.cor.2024.106665
Received 5 August 2023; Received in revised form 21 February 2024; Accepted 11 April 2024
Available online 18 April 2024

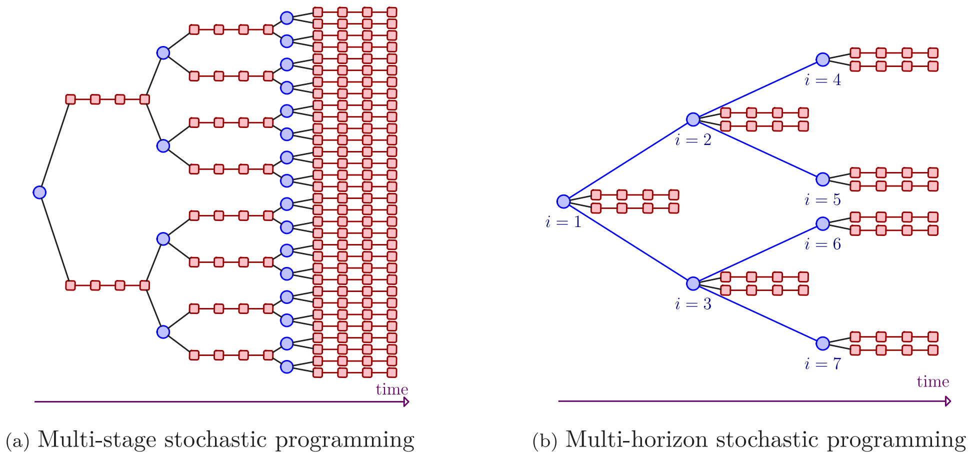


Fig. 1. Comparison between multi-stage stochastic programming and multi-horizon stochastic programming.

et al. (2020) proposed two inexact oracles that approximate $g(\mathbf{x}_i, \mathbf{c}_i)$ from below and above adaptively, and by using these, one can avoid solving all **SP**s every iteration, which reduces the computational cost compared with standard Benders decomposition. The method is called Adaptive Benders (AB). Like other Benders-type decomposition, AB can suffer from oscillation when solving degenerate models, which results in slower performance. Degeneracy can lead to oscillation and a severe slowdown in performance. Therefore, we address the degeneracy issue in this paper by developing a stabilised Benders decomposition with adaptive oracles. We call the improved method Stabilised Adaptive Benders (SAB) in the rest of the paper. The SAB consists of a level method stabilisation and a mechanism that dynamically selects the **SP**s to solve at every iteration. A good example of degeneracy is a multi-region power system planning problem. The main challenge of a multi-region problem compared with a single-region problem is that more regions lead to higher degeneracy, and Benders decomposition suffers from oscillation.

An important class of problems formulated in Eqs. (1)–(2) is multi-stage stochastic programming with short-term and long-term uncertainty. Including uncertainty from both time horizons using multi-stage stochastic programming may lead to a large scenario tree and an intractable model. Long-term infrastructure investment planning often faces uncertainty on two time horizons (Kaut et al., 2014; Lara et al., 2020): (a) the uncertainty from the operational time horizon and (b) the uncertainty from the strategic time horizon. Stochastic programming is often used to model uncertainty. Most studies on investment planning in a multi-horizon framework, such as Backe et al. (2022), only consider short-term uncertainty and treat investment stages as deterministic, and this helps to keep the problems tractable. However, long-term uncertainty can play an important role in investment planning. Although there are examples including short-term and long-term uncertainty in a multi-horizon model (Hellemo et al., 2013), the computational difficulty is not sufficiently addressed. Therefore, we aim to address the computational difficulties of long-term planning problems with short-term and long-term uncertainty. One possible way to reduce the problem size is to partly decouple the long- and short-term uncertainties by following the multi-horizon approach to stochastic programming (Kaut et al., 2014). A comparison example between multi-stage stochastic programming and multi-horizon stochastic programming is presented in Fig. 1. Although multi-horizon stochastic programming can reduce the size of the scenario tree significantly, it is essentially still a multi-stage stochastic programming when both short-term and long-term uncertainty are added and this can be intractable when the problem gets large. Another way is to develop an algorithm that can efficiently solve a class of large-scale optimisation problems, such as the progressive hedging type method (Munoz and Watson, 2015). Although some decomposition algorithms have been proposed to tackle the computational difficulty and claimed to be capable of solving problems with short-term and long-term uncertainty (Downward et al., 2020), the algorithms were only demonstrated to solve a problem with only short-term (Munoz et al., 2016) or long-term uncertainty (Singh et al., 2009). Therefore, this paper applies SAB to solve a long-term investment planning problem with short-term and long-term uncertainty.

We use the algorithm to solve an investment planning problem with short-term and long-term uncertainty along with exogenous quantities specific to the subproblem formulated as multi-horizon stochastic programming. In such a problem, $\mathbf{x}$, in Eqs. (1)–(2), represents investment decisions with corresponding expected investment cost $f(\mathbf{x})$. The investments affect a set $\mathcal{I}$ of operational nodes, and $\mathbf{x}_i$ formed from the subvector of $\mathbf{x}$ that represents the investments that affect node i, $\mathbf{c}_i$ specifies the operational costs, $\mathbf{y}_i$ defines the operational decisions at period i, and $g(\mathbf{x}_i, \mathbf{c}_i)$ gives the expected operational cost. The π_i is the probability associated with decision node i.

An illustration of a multi-horizon stochastic programming problem is presented in Fig. 1(b). The blue nodes represent the long-term uncertainty, and red squares represent the short-term uncertainty. The structure of multi-horizon stochastic programming allows us to apply Benders decomposition to solve multi-stage stochastic programming, which traditionally was handled by nested Benders decomposition (Birge, 1985). In traditional multi-stage stochastic programming, as shown in Fig. 1(a), the scenario tree is branched based on both long-term and short-term uncertainty, which leads to a large scenario tree. Also, the short-term node links the long-term node and vice versa, which requires nested Benders to decompose the problem. On the contrary, in multi-horizon stochastic programming, the short-term nodes are embedded in their corresponding long-term node, making the problem block separable and it is possible to directly apply Benders decomposition. In this case, in the Benders decomposition, all the long-term nodes (blue circles) are in the master problem, and all the short-term nodes are in the subproblems. Each block of short-term nodes embedded in their corresponding long-term node is a subproblem. Including long-term uncertainty makes the problem essentially a multi-stage stochastic programming problem. This difficulty is handled by applying Benders decomposition directly because of the structure of multi-horizon stochastic programming. The inclusion of short-term uncertainty leads to large subproblems. However, the subproblem, formulated as Eq. (2), inherently has the properties needed to utilise the adaptive oracles. Therefore, the difficulty of solving large subproblems is handled by utilising adaptive oracles in the solution process. Note that the short-term uncertainty embedded in the long-term node can also be revealed multiple times, which leads the subproblem to a multi-stage stochastic programming problem itself, but in this paper, we consider the case where short-term uncertainty is revealed only once in the corresponding long-term node.

The contributions of this paper are: (1) we develop SAB, a level method stabilisation of AB to address the oscillation issue and analyse the tuning of parameters; and (2) we test the proposed method on a multi-horizon stochastic programming model with short-term and long-term uncertainty and with an annual operational model with half hourly resolution. This results in problems with up to 1 billion variables and 4.5 billion constraints. The results show that it is up to 113.7 times faster than standard Benders and 2.1 times faster than the AB for a 1.00% convergence, and up to 45.5 times faster than standard Benders for a 0.10% convergence and the unstabilised AB cannot solve the largest instance to convergence tolerance; (3) we propose dynamic level method stabilisation to increase the robustness of the proposed method and can be up to 33.5 times faster for 1.00% convergence and 25.4 times faster for 0.10% convergence compared with standard level method stabilisation with poor parameter choices.

The outline of the paper is as follows: Section 2 introduces the background knowledge regarding stochastic programming, multi-horizon modelling approach, Benders decomposition and stabilisation. Section 3 introduces the level method stabilisation. Section 4 gives the problem description. Section 5 presents the model for the case study. Section 6 states the computational results and numerical analysis. Section 7 discusses the implications of the method and results and summarises the limitations of the research. Section 8 concludes the paper and suggests further research.

2. Literature review

This paper proposes a Benders-type algorithm to solve large-scale optimisation problems. In the following, we present the background knowledge of stochastic programming, multi-horizon modelling approach, standard Benders decomposition, AB, and level method stabilisation.

2.1. Stochastic programming

Stochastic programming is part of mathematical programming and operations research that studies how to incorporate uncertainty into decision problems (King and Wallace, 2012). It is one of the most popular methods of dealing with uncertainties in energy system planning (Birge and Louveaux, 2011). The electricity system in regulated markets is a well-developed area for using stochastic programming in energy (Wallace and Fleten, 2003; Powell and Meisel, 2016). However, stochastic programming is also exploited in natural gas systems (Fodstad et al., 2016), offshore oil and gas infrastructure planning (Gupta and Grossmann, 2014), and hydrogen network (Galan et al., 2019).

Two-stage stochastic programming (Boffino et al., 2019), multi-stage stochastic programming (Pereira and Pinto, 1991), stochastic mixed-integer programming (Salo et al., 2022; Lara et al., 2020; Munoz et al., 2016), and stochastic nonlinear programming (Li, 2021) are all used in energy system research. In Lara et al. (2020), a multi-stage stochastic mixed-integer programming formulation was developed to optimise electricity infrastructure planning over multiple years. To solve a large-scale model, they decomposed and solved the problem using parallelised stochastic dual dynamic integer programming.

2.2. Multi-horizon stochastic programming

In traditional multi-stage stochastic programming, uncertainty in both the operational and strategic time horizons can lead to a huge scenario tree, thus, an intractable problem. The multi-horizon modelling approach was proposed as an alternative formulation that reduces the model size significantly (Kaut et al., 2014). One can achieve a much smaller model by disconnecting operational problems from the following strategic nodes. The resulting model is called multi-horizon stochastic programming. An illustrative example of multi-horizon stochastic programming is presented in Fig. 1(b). The multi-horizon formulation is an approximation to multi-stage stochastic programming, but the approximation will be exact enough provided the state of the operational system at the end of a stage has no significant influence on the optimal strategic decisions in the later stages. In the examples in this paper, this corresponds to being able to ignore the reservoir levels at the end of a stage when deciding the future investment.

2.3. Benders decomposition

Benders decomposition was first developed in Benders (1962) and has been successfully applied to a wide range of difficult optimisation problems (Rahmaniani et al., 2017). Benders decomposition exploits the block diagonal structure of Eqs. (1)–(2) and creates outer linearisation. This method has been used in stochastic programming and is known as the L-shaped method (Slyke and Wets, 1969). Directly solving (1) –(2) can be prohibitive if there are a large number of decision nodes, which occurs in stochastic programming when there are multiple stages or many uncertain parameters.

In Benders decomposition, instead of solving Eqs. (1) –(2) directly, a sequence of approximations is solved. Two types of constraints can be added after each solve: feasibility cuts (enforcing the feasibility of (1)) and optimality cuts (linear approximations to (1)) (Birge and Louveaux, 2011). The standard Benders decomposition is presented in Algorithm 1. At iteration j, the Relaxed Master Problem (**RMP**) is

$$\min_{\mathbf{x}\in\mathcal{X},\beta} f(\mathbf{x}) + \sum_{i\in\mathcal{I}} \pi_i \beta_i \tag{3a}$$

$$\text{s.t.} \quad \beta_i \geq \theta + \lambda^\top(\mathbf{x}_i - \mathbf{x}), \qquad (\mathbf{x},\theta,\lambda) \in \mathcal{F}_{i(j-1)}, i \in \mathcal{I}, \tag{3b}$$

where $\mathcal{F}_{i(j-1)}$ is the set of cuts associated with **SP** i generated prior to iteration j. In iteration j of Benders decomposition, we first solve the **RMP** to obtain a solution $\mathbf{x}_j$. Then we extract the subvector $\mathbf{x}_{ij}$ of $\mathbf{x}_j$, corresponding to **SP** i as its right hand side parameters. Solving this gives the optimal value of the **SP**, θ_{ij}, and a subgradient, λ_{ij} at $\mathbf{x}_{ij}$. Finally, new cutting planes are added to $\mathcal{F}_{i(j-1)}$ which give $\mathcal{F}_{ij} := \mathcal{F}_{i(j-1)} \cup \{\mathbf{x}_{ij}, \theta_{ij}, \lambda_{ij}\}$. This version is referred to as multi-cut Benders decomposition. The algorithm iterates until the upper bound and the lower bound converge. The lower bound is the optimal value of the **RMP**. The upper bound is the best feasible solution. Benders decomposition will converge in a finite number of iterations when the **SP** is linear programming.

2.4. Benders decomposition with adaptive oracles

Benders-type algorithms iteratively approximate the **SP** cost function through a set of cutting planes. However, Benders decomposition may get slow severely when there are many **SP**s. Therefore, research on making Benders decomposition more efficient was conducted (Skar et al., 2014; Zakeri et al., 2000; Baena et al., 2020). One approach is to exploit the **SP** structure to avoid solving all **SP**s but still get a valid cutting plane at each iteration.

When $g(\mathbf{x}_i, \mathbf{c}_i)$ is convex and decreasing w.r.t. $\mathbf{x}_i$, and concave and increasing w.r.t. $\mathbf{c}_i$, one can exploit these properties to solve the problem more efficiently (Mazzi et al., 2020). Mazzi et al. (2020) proposed two inexact oracles that approximate $g(\mathbf{x}_i, \mathbf{c}_i)$ from below and above adaptively, and by using these, one can avoid solving all **SP**s every iteration, and so reduce the computational cost compared to standard Benders decomposition. The **RMP** in AB has the same form as in standard Benders, but with a different set of cuts $\mathcal{F}_j$ given by the inexact adaptive oracles. The adaptive oracles provide inexact but valid upper and lower bounds on θ, $\overline{\theta}$ and $\underline{\theta}$, and give a value $\underline{\lambda}$ of λ which yields a valid cutting plane.

Like other Benders-type decomposition, AB suffers from oscillation and may result in slower performance. The performance of the AB method was tested on a UK power system planning problem (Mazzi

Algorithm 1 Standard Benders

choose ϵ (convergence tolerance), $\underline{\beta}$ (initial lower bound for β_i), $U_0^* := M$ (initial upper bound), set $j := 0$, $\mathcal{F}_{i0} := \{(\beta_{i0}, 0, 0)\}$ for each $i \in \mathcal{I}$;
repeat
  $j := j + 1$;
  solve **RMP** and obtain β_{ij} and $\mathbf{x}_j^{RMP}$; set $L_j^* := f(\mathbf{x}_j^{RMP}) + \sum_{i\in\mathcal{I}} \pi_i \beta_{ij}$;
  for $i \in \mathcal{I}$ **do**
    solve **SP** i at $(\mathbf{x}_{ij}^{RMP}, \mathbf{c}_i)$ and obtain θ_{ij} and λ_{ij};
  end for
  for $i \in \mathcal{I}$ **do**
    $\mathcal{F}_{ij} := \mathcal{F}_{i(j-1)} \cup \{(\mathbf{x}_{ij}^{RMP}, \theta_{ij}, \lambda_{ij})\}$;
  end for
  $U_j^* := \min(U_{j-1}^*, f(\mathbf{x}_j^{RMP}) + \sum_{i\in\mathcal{I}} \pi_i \theta_{ij})$;
until $U_j^* - L_j^* \leq \epsilon$.

et al., 2020). However, it is a single-region investment planning problem, and we found that the algorithm becomes slower after introducing more regions into the problem. This issue must be addressed because a power system investment planning problem normally involves multiple regions connected via transmission lines (Gacitua et al., 2018). Therefore, this paper develops a stabilised Benders decomposition with adaptive oracles. We call the improved method SAB in the rest of the paper. The SAB consist of a level method stabilisation and a mechanism that dynamically selects the **SP**s to solve at every iteration.

3. Level method stabilisation

In this paper, based on the observation that Benders-type algorithms suffer from severe oscillation when solving multi-region planning problems, we stabilise the algorithm in Mazzi et al. (2020) using the level method. The level method was introduced in Lemarechal et al. (1995). It was then used to regularise standard Benders decomposition (Fabian, 2000).

We now present the stabilisation step and its coordination with AB. At each iteration j, the Level Method Problem (**LMP**) for stabilisation can be formulated as

$$\mathbf{x}_j^{LMP} = \underset{\mathbf{x}\in\mathcal{X},\beta}{\operatorname{argmin}} \left\| \mathbf{x} - \mathbf{x}_{j-1}^{LMP} \right\|_2 \tag{4a}$$

$$\text{s.t.} \quad \beta_i \geq \theta + \lambda^\top (\mathbf{x}_i - \mathbf{x}), \qquad (\mathbf{x}, \theta, \lambda) \in \mathcal{F}_{i(j-1)}, i \in \mathcal{I}, \tag{4b}$$

$$f(\mathbf{x}) + \sum_{i\in\mathcal{I}} \pi_i \beta_i \leq T_j, \tag{4c}$$

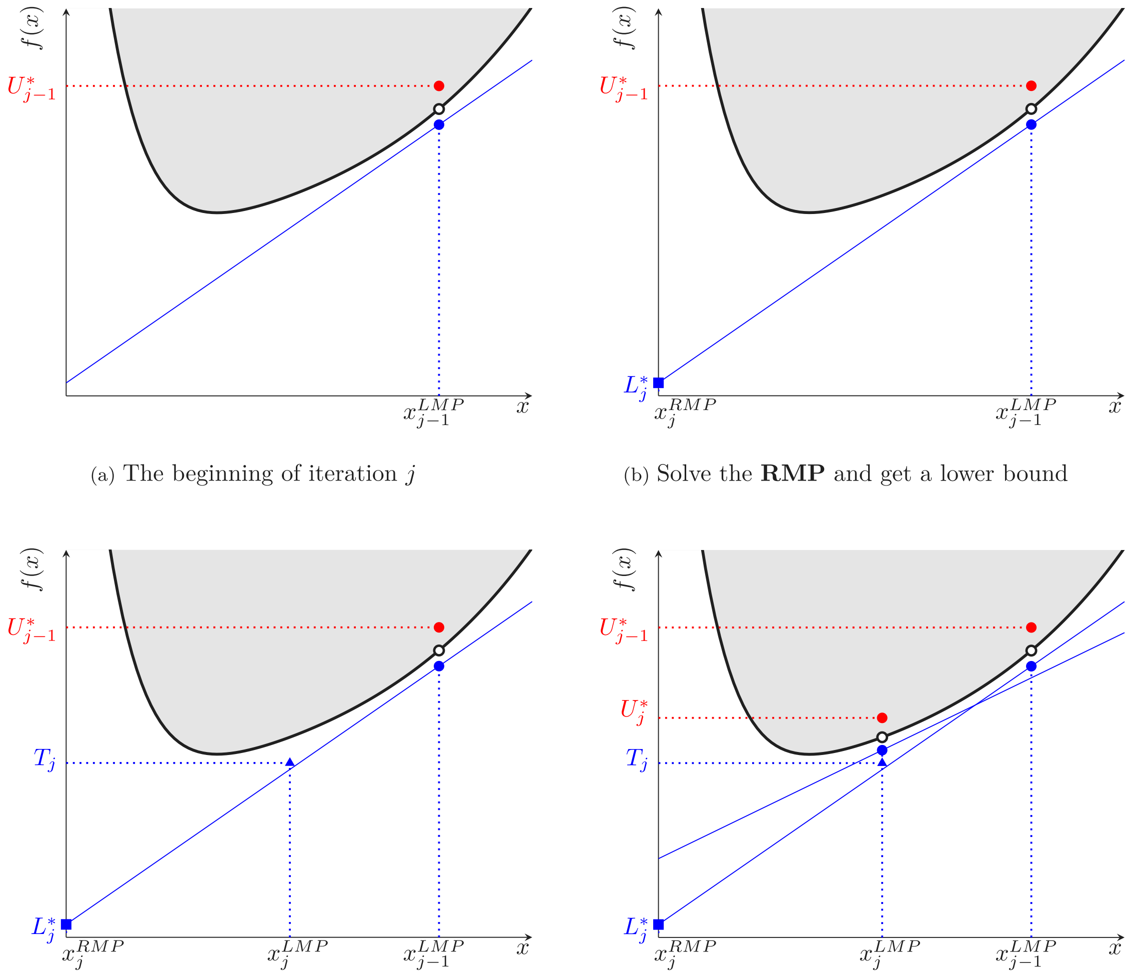
Constraint (4c) is the level set, and $T_j = L_j^* + \gamma(U_{j-1}^* - L_j^*)$ is the target **RMP** objective value, where L_j^* is the lower bound found in iteration j and U_{j-1}^* is the upper bound found in iteration $j-1$. The stabilisation factor, γ, can be interpreted as the ratio of the new targeted bound gap to the existing bound gap. **LMP** minimises the distance from the previous reference point subject to all the constraints from the **RMP** and an extra target constraint enforcing a maximum allowed level for the **RMP** objective.

A graphical illustration of the level method SAB decomposition is presented in Fig. 2. At the beginning of iteration j, we have the cuts that have been added in all previous iterations, the upper bound U_{j-1}^* and the lower approximation of the function value (blue dot). The black dot represents the function value at that point which is unknown unless all **SP**s are solved exactly. The **RMP** is resolved and a new lower bound L_j^* (blue square) is obtained. If there is no stabilisation, then the solution x_j^{RMP} will be the next sampled point at which the **SP**s are evaluated. When there is stabilisation, the moving area is restricted, and we move to the closest point to x_{j-1}^{LMP} that does not exceed the target T_j. At point x_j^{LMP}, we evaluate one or more **SP**s and get a new upper bound U_j^* (red dot) and add cuts.

3.1. Stabilised benders decomposition with adaptive oracles algorithm

In this section, we present the stabilised Benders decomposition with the adaptive oracles algorithm, shown in Algorithm 2. We initialise Algorithm 2 by choosing convergence tolerance, stabilisation factor $\gamma \in (0, 1)$, and initial lower and upper bounds. If dynamic level set is used, then we initialise $\omega \in (0, 1)$, a constant that increases or decreases γ and thresholds, $\overline{P}$ and $\underline{P}$ ($0 \leq \underline{P} < \overline{P} \leq 1$), which determine whether to increase or decrease γ. Next, we solve a **SP** at the special point (Line 3). This point is used by the inexact oracles, and we refer the readers to Mazzi et al. (2020) for details. In the first iteration, a **RMP** is solved, and we use the solution to the **RMP** as the reference point for the stabilisation for the second iteration. At the newly proposed point, we call oracles for each node (Line 13–15). Then we adaptively select and solve **SP**s (Line 16–25). In Line 17, we choose the **SP** that has the largest contribution to the gap and solve it exactly. The valid upper and lower bounds for the **SP** at node i are denoted by $\overline{\theta}_{ij}$ and $\underline{\theta}_{ij}$. After a **SP** is solved exactly, the information for the oracles is updated in Line 19, where ϕ_{ij} is the sensitivity to the cost coefficients and is used in evaluating adaptive oracles for other subproblems. In Line 20–22, the adaptive oracles are called again because more information has been added to the oracles. Line 25 states the stopping rule for evaluating **SP**s, $U_j^{UBO} - L_j^{LBO} \leq U_{j-1}^* - L_{j-i}^*$ means locally improved, and $L_j^{LBO} \geq U_{j-1}^*$ means that the lower bound at the current point is higher than the upper bound of the problem, which means that the current point cannot improve the upper bound so we should not evaluate more **SP**s. Then in Line 26–28, one cut is added for each subproblem i. Lines 30–39 are the level set management steps, which are explained in the following. The algorithm terminates in Line 40 if it reaches the convergence tolerance.

Although a common feature of the level method compared with other bundle-type methods is that the parameters do not need to be adjusted (Zverovich et al., 2012), nevertheless, there are occasions when it may be beneficial. Therefore, in addition to using fixed stabilisation factor γ, we also explore methods for adjusting the stabilisation factor γ with the aim of achieving a more robust algorithm. There are several ways to adjust stabilisation dynamically, and we choose a method analogous to what is used to adjust trust regions. The trust region method uses a local approximation of the function to be minimised and optimised within the trust region. The trust region size is updated throughout the iterations. In the trust region method, one adjusts the trust region according to the ratio of the actual decrease to the predicted decrease (Fletcher, 2000). Inspired by Fletcher (2000), we adjust the level set based on the ratio, r, of the actual improvement to the expected improvement, where I_j^A is the actual improvement from iteration $j-1$ to j, and I_j^P is the predicted improvement from iteration $j-1$ to j. Then we update γ. If both the actual improvement and predicted improvement are positive (Line 32), we check the ratio against two predefined parameters $\underline{P}$ and $\overline{P}$. If the improvement ratio r is lower than $\underline{P}$, meaning that the actual improvement is less than the least improvement we want to achieve in proportion to the



(a) The beginning of iteration j

(b) Solve the **RMP** and get a lower bound

(c) Set the target and solve the **LMP**

(d) Get an upper bound and add a cut

Fig. 2. An illustrative example from iteration $j-1$ to iteration j.

predicted improvement, we tighten the stabilisation in Line 34. This is because we want to avoid moving to an area that provides insufficient improvement. The parameter ω affects how significant the change in stabilisation is. If the improvement ratio is higher than $\overline{P}$, the actual improvement aligns with the predicted improvement well. This suggests we are moving in the right direction, and we loosen the stabilisation to hopefully achieve more improvement in the next iteration.

Unlike standard Benders where the exact value of the **SP**s is known, in SAB only lower and upper bounds on the objective values are known. By comparing the lower and upper bounds with the exact values of the **SP**s, we find that the lower bound oracle gives a much closer and more stable approximation. Therefore, we use L^{LBO} instead of U^{UBO} when defining the ratio r. Furthermore, a bad approximation from the lower bound oracle at the current point or a bad approximation from the upper bound oracle from previous points can lead to a negative I^P_j. Therefore, we choose to do nothing if I^P_j is negative. For I^A_j, there are two possibilities for it to be negative: (1) bad approximation from the lower bound oracle or the upper bound oracle, and (2) going to a bad sampled point. A bad sampled point, in this case, means the L^{LBO} at the current iteration is higher than the L^{LBO} at the previous iteration. If the information is exact and I^A_j is negative, one may reject the point, go back to the best point seen so far, and try again with a higher γ. However, in the case of inexact information, it may not be sensible to reject a point based on a bad approximation. In the computational study in this paper, both dynamic and fixed stabilisation are used.

4. Problem description, modelling strategies and modelling assumptions

The examples used to test the algorithm are designed to choose the cost optimal investment strategy and operational scheduling for a power system to achieve emission targets. In this section, we present the problem description, temporal, transmission and geographical representations of the problem and the modelling assumptions.

The problem under consideration aims to make optimal multi-period investment planning regarding capacity expansion for both generation and transmission and operational decisions for the UK power system that satisfies the emission reduction goals under (a) short-term uncertainty, including renewable energy availability and load profile; and (b) long-term uncertainty, including CO_2 budget, CO_2 tax, and long-term power demand.

For the investment planning, we consider: (a) thermal generators (Coal-fired plant, OCGT, CCGT, Diesel, and nuclear plants); (b) generators with Carbon Capture and Storage (CCS) (Coal-fired plant with CCS); (c) renewable generators (offshore wind, onshore wind and solar PV); (d) electric storage (PHES and lithium); and (e) transmission lines. All the technologies have some historical capacities, and the model aims to make multi-period investments planning in the optimal mix of technologies to meet future power demand and emission targets. The capital expenditures and fixed operational costs are assumed to be known.

Algorithm 2 Level method stabilised Benders decomposition with adaptive oracles

choose ϵ (convergence tolerance), γ (stabilisation factor), $\underline{\beta}$ (initial lower bound β_i), $U_0^* := M$ (initial upper bound), $\omega \in (0,1)$, $\underline{P} \in (0,1)$, and $\overline{P} \in (\underline{P}, 1)$;
set $j := 0$, $\mathcal{F}_{i0} := \{(\beta_{i0}, 0, 0)\}$ for each $i \in \mathcal{I}$;
solve **SP** at the special point $(\underline{\mathbf{x}}, \underline{\mathbf{c}})$ and obtain θ, λ and $\boldsymbol{\phi}$; $\mathcal{S} := \{(\underline{\mathbf{x}}, \underline{\mathbf{c}}, \theta, \lambda, \boldsymbol{\phi})\}$;
repeat
  $j := j + 1$;
  solve **RMP** and obtain β_{ij} and $\mathbf{x}_j^{RMP}$; set $L_j^* := f(\mathbf{x}_j^{RMP}) + \sum_{i\in\mathcal{I}} \pi_i \beta_{ij}$;
  if $j = 1$ **then**
    $\mathbf{x}_j^{LMP} := \mathbf{x}_j^{RMP}$;
  else
    $T_j := L_j^* - \gamma \left(U_{j-1}^* - L_j^*\right)$;
    solve **LMP** and obtain $\mathbf{x}_j^{LMP}$;
  end if
  for $i \in \mathcal{I}$ **do**
    call adaptive oracles at $(\mathbf{x}_{ij}^{LMP}, \mathbf{c}_i)$ and obtain $\underline{\theta}_{ij}$, $\overline{\theta}_{ij}$, and $\underline{\lambda}_{ij}$;
  end for
  repeat
    $i := \mathrm{argmax}_{i\in\mathcal{I}}\, \pi_i(\overline{\theta}_{ij} - \underline{\theta}_{ij})$;
    solve **SP**$_i$ at $(\mathbf{x}_{ij}^{LMP}, \mathbf{c}_i)$ exactly and obtain θ_{ij}, λ_{ij}, $\boldsymbol{\phi}_{ij}$;
    $\mathcal{S} := \mathcal{S} \cup \{(\mathbf{x}_{ij}^{LMP}, \mathbf{c}_i, \theta_{ij}, \lambda_{ij}, \boldsymbol{\phi}_{ij})\}$;
    for $i \in \mathcal{I}$ **do**
      call adaptive oracles at $(\mathbf{x}_{ij}^{LMP}, \mathbf{c}_i)$ and obtain $\underline{\theta}_{ij}$, $\overline{\theta}_{ij}$, and $\underline{\lambda}_{ij}$;
    end for
    $L_j^{LBO} := f(\mathbf{x}_j^{LMP}) + \sum_{i\in\mathcal{I}} \pi_i \underline{\theta}_{ij}$;
    $U_j^{UBO} := f(\mathbf{x}_j^{LMP}) + \sum_{i\in\mathcal{I}} \pi_i \overline{\theta}_{ij}$;
  until $U_j^{UBO} - L_j^{LBO} \le U_{j-1}^* - L_{j-1}^*$ **or** $L_j^{LBO} \ge U_{j-1}^*$;
  for $i \in \mathcal{I}$ **do**
    $\mathcal{F}_{ij} := \mathcal{F}_{i(j-1)} \cup \{(\mathbf{x}_{ij}^{LMP}, \underline{\theta}_{ij}, \underline{\lambda}_{ij})\}$;
  end for
  $U_j^* := \min(U_{j-1}^*, U_j^{UBO})$;
  if dynamic level set **then**
    $I_j^P := L_{j-1}^{LBO} - T_j$, $I_j^A := L_{j-1}^{LBO} - L_j^{LBO}$, $r := \frac{I_j^A}{I_j^P}$;
    if $I_j^A > 0$ **and** $I_j^P > 0$ **then**
      if $r \le \underline{P}$ **then**
        $\gamma := 1 - \omega(1 - \gamma)$;
      else if $r > \overline{P}$ **then**
        $\gamma := \omega\gamma$;
      end if
    end if
  end if
until $U_j^* - L_j^* \le \epsilon$.

For the operational part, variable operational costs, fuel costs, efficiency and ramping limits of the thermal generators are known. The variable costs and charging efficiency of the electric storage are known.

The problem is to determine: (a) the capacities and timing of invested technologies and (b) operational strategies that include scheduling of generators, storage and approximate power flow among regions to meet the power demand every half hour with minimum overall investment, operational and environmental costs across the planning horizon.

4.1. Modelling strategies and assumptions

In this section, we present the modelling strategies and assumptions we use in the stochastic long-term multi-region multi-period investment planning problem.

4.2. Temporal representation

Long-term planning problems can involve decades of time horizon. Solving the long-term planning problem with operational decisions in every hour of the planning horizon was considered intractable (Li et al., 2022). Several works propose to select representative days to represent the hourly fluctuations. In this problem, we consider operational decisions every half hour in a year.

4.2.1. Scenario generation

For short-term uncertainty, we select time intervals from historical data that represent the different demand and weather conditions over the year to represent a whole year in an operational scenario. An illustration of an operational scenario is presented in Fig. 3.

For long-term uncertainty, each independent uncertain parameter has n possible outcomes in the next stage, each of which is linked to further n possible outcomes in the following stage. The realisations in one stage are assigned with an equal probability. We use a reasonably

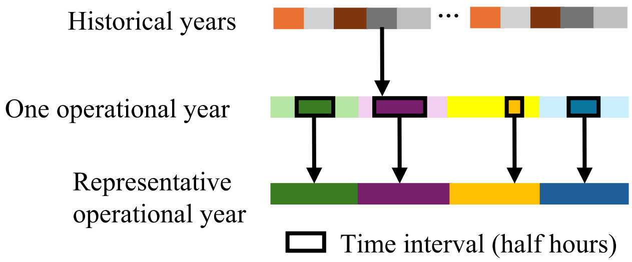


Fig. 3. An illustrative example of an operational scenario.

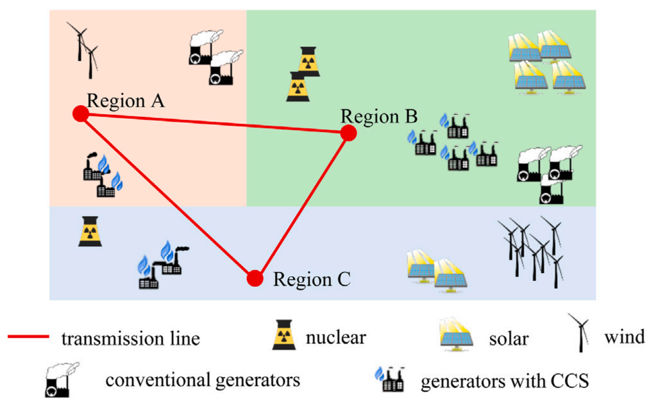


Fig. 4. An illustrative example of the planning problem.
Source: Adapted from Lara et al. (2020).

simple scenario generation routine because scenario generation is not the focus of the paper, and we refer the readers to King and Wallace (2012) and Fairbrother et al. (2022) for more advanced scenario generation approaches.

4.3. Transmission representation

We consider existing transmission lines and expand capacity to them. The length and capacity of each transmission line are known. For the existing transmission lines, we assume that they will not reach their life expectancy during the planning horizon, i.e., we do not consider the retirement of transmission lines. For the candidate transmission lines, the capital cost of each transmission line is known.

4.3.1. Geographical representation

The problem potentially consists of many regions and results in a large model. Therefore, we aggregate regions into representative ones to reduce the number of locations. The generators and storage units in one region with the same characteristics are aggregated into clusters. In such a way, the model does not invest in a specific unit but in that type of device, and a linear investment model may be sufficient in this case. An illustration of a simple example is presented in Fig. 4.

4.3.2. Modelling assumptions

We assume that: (a) a linear cost model for each technology because we deal with an aggregated system and the fixed part of the investment cost can be evened out and lead to a linear programming master problem; (b) the Kirchhoff voltage law is omitted; (c) no loss in the transmission lines, and (d) the initial energy storage level of storage facilities are half of their capacities.

5. Mathematical model

This section presents the mathematical model for the power system investment planning and operational optimisation problem. The problem is decomposed by having an investment planning master problem and an operational **SP**. The complete nomenclature of the model can be found in Section 5.1. We use the conventions that calligraphic capitalised Roman letters denote sets, upper case Roman and lower case Greek letters denote parameters, and lower case Roman letters denote variables. The indices are subscripts, and name extensions are superscripts. The same lead symbol represents the same type of thing. The names of variables, parameters, sets and indices are single symbols.

5.1. Nomenclature

Investment planning model sets	
$\mathcal{P}$	Set of technologies, p
$\mathcal{I}$	Set of operational nodes, i
$\mathcal{I}_0$	Set of investment nodes, i_0
$\mathcal{I}_i$	Set of investment nodes i_0 ($i_0 \in \mathcal{I}_0$) ancestor to operational node i ($i \in \mathcal{I}$)
Operational model sets	
$\mathcal{N}$	Set of time slices, n
$\mathcal{T}$	Set of hours in all time slices, t
$\mathcal{T}_n$	Set of hours in time slice n, t
$\mathcal{L}$	Set of transmission lines, l
$\mathcal{Z}$	Set of regions
$\mathcal{L}_z^{Out/In}$	Set of transmission lines go out of/into region z, l
$\mathcal{G}/\mathcal{G}_z$	Set of all thermal generators/in region z, g
$\mathcal{S}/\mathcal{S}_z$	Set of all electricity storage/in region z, s
$\mathcal{R}/\mathcal{R}_z$	Set of all renewable generations/in region z, r
Investment planning model parameters	
C_{pi}^{Inv}	Unitary investment cost of device p in investment node i ($p \in \mathcal{P}, i \in \mathcal{I}_0$) [£/MW]
C_{pi}^{Fix}	Unitary fix operational and maintenance cost of device p ($p \in \mathcal{P}, i \in \mathcal{I}$) [£/MW]
X_{pi}^{Init}	Initial capacity of device p ($p \in \mathcal{P}, i \in \mathcal{I}$) [MW]
X_p^{Max}	Maximum installed capacity of device p ($p \in \mathcal{P}$) [MW]
κ	Scaling effect depending on the number of operation years between investment nodes
$\delta_i^{I_0}/\delta_i^{I}$	Discount factor of investment node i ($i_0 \in \mathcal{I}_0$)/operational node i ($i \in \mathcal{I}$)
$\pi_i^{I_0}/\pi_i^{I}$	Probability of investment node i ($i_0 \in \mathcal{I}_0$)/operational node i ($i \in \mathcal{I}$)
H_p^P	Life time of technology p ($p \in \mathcal{P}$)
$\mathbf{x}_i$	Right hand side coefficients of the operational subproblem
$\mathbf{c}_i$	Cost coefficients of the operational subproblem
μ_i^E	CO_2 budget at operational node i ($i \in \mathcal{I}$)
μ_i^{DP}	Scaling factor on power demand at operational node i ($i \in \mathcal{I}$)
I_i^L	Strategic stage of node i ($i \in \mathcal{I} \cup \mathcal{I}_0$)
C_i^{CO2}	CO_2 emission price at operational node i ($i \in \mathcal{I}$)
Operational model parameters	
W_t	Weight of operational period t ($t \in \mathcal{T}$)
H_t	Number of hour(s) in one operational period t ($t \in \mathcal{T}$)
α_g^G	Maximum ramp rate of gas turbines ($g \in \mathcal{G}$) [MW/MW]

R^R_{rt}	Capacity factor of renewable unit r in period t ($r \in \mathcal{R}, t \in \mathcal{T}$)
η^{SE}_s	Efficiency of electricity store s ($s \in \mathcal{S}$)
γ^{SE}_s	Power ratio of electricity store s ($s \in \mathcal{S}$) [MWh/MW]
E^G_g	Emission factor of gas turbine g ($g \in \mathcal{G}$) [tonne/MWh]
C^G_g/C^{SE}_s	Operational unit cost of a generator g/a storage facility s ($g \in \mathcal{G}/s \in \mathcal{S}$) [£/MW]
C^{ShedP}	Load shed penalty cost [£/MWh]
P^{DP}_{zt}	Power demand at region z period t ($z \in \mathcal{Z}, t \in \mathcal{T}$) [MW]
Investment planning model variables	
x^{Acc}_{pi}	Accumulated capacity of device p in operational node i ($p \in \mathcal{P}, i \in \mathcal{I}$) [MW]
x^{Inst}_{pi}	Newly invested capacity of device p in investment node i_0 ($p \in \mathcal{P}, i \in \mathcal{I}_0$) [MW]
c^{INV}	Total expected investment cost [£]
Operational model variables	
p^{AccG}_g	Accumulated capacity of generator g ($g \in \mathcal{G}$) [MW]
p^{AccL}_l	Accumulated capacity of line l ($l \in \mathcal{L}$) [MW]
p^{AccR}_g	Accumulated capacity of renewable unit r ($r \in \mathcal{R}$) [MW]
p^{AccSE}_s	Accumulated charging/discharging capacity of electricity store s ($s \in \mathcal{S}$) [MW]
p^G_{gt}	Power generation of generator g in period t ($g \in \mathcal{G}, t \in \mathcal{T}$) [MW]
$p^{SE+}_{st}/p^{SE-}_{st}$	Charge/discharge power of electricity store s in period t ($s \in \mathcal{S}, t \in \mathcal{T}$) [MW]
p^L_{lt}	Power flow in line l in period t ($l \in \mathcal{L}, t \in \mathcal{T}$) [MW]
p^{GShedP}_{zt}	Generation shed at z in period t ($z \in \mathcal{Z}, t \in \mathcal{T}$) [MW]
q^{SE}_{st}	Energy level of electricity store s at the start of period t ($s \in \mathcal{S}, t \in \mathcal{T}$) [MWh]
p^{ShedP}_{zt}	Load shed at z in period t ($z \in \mathcal{Z}, t \in \mathcal{T}$) [MW]
Function	
$c^{OPE}(\cdot,\cdot)$	Operational cost at operational node i ($i \in \mathcal{I}$) [£]

5.2. Investment planning model

$$\min \quad c^{INV} + \kappa \sum_{i\in\mathcal{I}} \pi^I_i c^{OPE}(\mathbf{x}_i, \mathbf{c}_i) \tag{5a}$$

$$\text{s.t.} \quad c^{INV} = \sum_{i\in\mathcal{I}_0} \delta^{I_0}_i \pi^{I_0}_i \sum_{p\in\mathcal{P}} C^{Inv}_{pi} x^{Inst}_{pi} + \kappa \sum_{i\in\mathcal{I}} \delta^I_i \pi^I_i \sum_{p\in\mathcal{P}} C^{Fix}_{pi} x^{Acc}_{pi}, \tag{5b}$$

$$x^{Acc}_{pi} = X^{Init}_{pi} + \sum_{i_0\in\mathcal{I}_i | \kappa(I^L_i - I^L_{i_0}) \le H^P_p} x^{Inst}_{pi}, \quad p \in \mathcal{P}, i \in \mathcal{I}, \tag{5c}$$

$$x^{Acc}_{pi} \le X^{Max}_p, \quad p \in \mathcal{P}, i \in \mathcal{I}, \tag{5d}$$

$$x^{Inst}_{pi}, x^{Acc}_{pi} \in \mathbb{R}^+_0. \tag{5e}$$

The total cost for investment planning, Eq. (5a), consists of actual discounted investment costs and discounted fixed operating and maintenance costs c^{INV}, as well as the expected operational cost of the system over the time horizon $\kappa \sum_{i\in\mathcal{I}} \pi^I_i c^{OPE}(\mathbf{x}_i, \mathbf{c}_i)$. Here, κ is a scaling factor that depends on the time step between two successive investment nodes. Constraint (5c) states that the accumulated capacity of a technology x^{Acc}_{pi} in an operational node equals the sum of the initial capacity X^{Init}_p and newly invested capacities x^{Inst}_{pi} in its ancestor investment nodes $\mathcal{I}_i$ that are in their lifetimes. The parameter X^{Max}_p denotes the maximum accumulated capacity of technologies. We define $\mathbf{x}_i = \left(\{x^{Acc}_{pi}, p \in \mathcal{P}\}, \mu^{DP}_i, \mu^E_i\right), i \in \mathcal{I}$ that collects all right hand side coefficients that will be fixed in the **SP**, Eqs. (6), into vector $\mathbf{x}_i$. The $\mathbf{c}_i = \left(C^{CO_2}_i\right), i \in \mathcal{I}$ collects all the cost coefficients into vector $\mathbf{c}_i$.

5.3. Operational model

We now compute the operational cost $c^{OPE}(\mathbf{x}_i, \mathbf{c}_i)$ at one operational node i by solving **SP**, Eqs. (6) given the decisions $\mathbf{x}_i$ and $\mathbf{c}_i$ made in the master problem, Eqs. (5). The function $c^{OPE}(\mathbf{x}_i, \mathbf{c}_i)$ corresponds to $g(\mathbf{x}_i, \mathbf{c}_i)$ in Eq. (2) Note that we omit index i in the operational model for ease of notation.

$$C^{OPE}\left(\{p^{AccG}_g\ g \in \mathcal{G}\}, \{p^{AccSE}_s, s \in \mathcal{S}\}, \{p^{AccL}_l, l \in \mathcal{L}\}, \mu^{DP}, \mu^E, C^{CO_2}\right) :=$$

$$\min \quad \sum_{t\in\mathcal{T}} W_t H_t \left(\sum_{g\in\mathcal{G}} C^G_g p^G_{gt} + \sum_{s\in\mathcal{S}} C^{SE}_s p^{SE+}_{st} + \sum_{z\in\mathcal{Z}} C^{ShedP} p^{ShedP}_{zt} \right) \tag{6a}$$

$$\text{s.t.} \quad p^G_{gt} \le p^{AccG}_g, \quad g \in \mathcal{G}, t \in \mathcal{T}, \tag{6b}$$

$$-p^{AccL}_l \le p^L_{lt} \le p^{AccL}_l, \quad l \in \mathcal{L}, t \in \mathcal{T}, \tag{6c}$$

$$p^{SE+}_{st} \le p^{AccSE}_s, \quad s \in \mathcal{S}, t \in \mathcal{T}, \tag{6d}$$

$$p^{SE-}_{st} \le p^{AccSE}_s, \quad s \in \mathcal{S}, t \in \mathcal{T}, \tag{6e}$$

$$q^{SE}_{st} \le \gamma^{SE}_s p^{AccSE}_s, \quad s \in \mathcal{S}, t \in \mathcal{T}, \tag{6f}$$

$$-\alpha^G_g H_t p^{AccG}_g \le p^G_{gt} - p^G_{g(t-1)} \le \alpha^G_g H_t p^{AccG}_g, \quad g \in \mathcal{G}, n \in \mathcal{N}, t \in \mathcal{T}_n, \tag{6g}$$

$$\sum_{g\in\mathcal{G}_z} p^G_{gt} + \sum_{l\in\mathcal{L}^{In}_z} p^L_{lt} + \sum_{s\in\mathcal{S}_z} p^{SE-}_{st} + \sum_{r\in\mathcal{R}_z} R^R_{rt} p^{AccR}_r + p^{ShedP}_{zt} = \mu^{DP} P^{DP}_{zt} + \sum_{l\in\mathcal{L}^{Out}_z} p^L_{lt} + \sum_{s\in\mathcal{S}_z} p^{SE+}_{st} + p^{GShedP}_{zt}, \quad z \in \mathcal{Z}, t \in \mathcal{T}, \tag{6h}$$

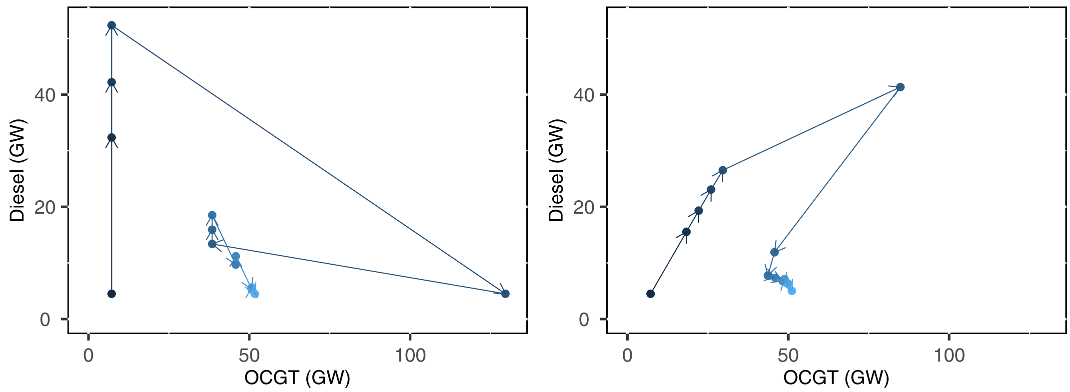
$$q^{SE}_{s(t+1)} = q^{SE}_{st} + H_t(\eta^{SE}_s p^{SE+}_{st} - p^{SE-}_{st}), \quad s \in \mathcal{S}, n \in \mathcal{N}, t \in \mathcal{T}_n, \tag{6i}$$

$$\sum_{t\in\mathcal{T}} \sum_{g\in\mathcal{G}} W_t H_t E^G_g p^G_{gt} \le \mu^E, \tag{6j}$$

$$p^L_{lt} \in \mathbb{R}_0, \tag{6k}$$

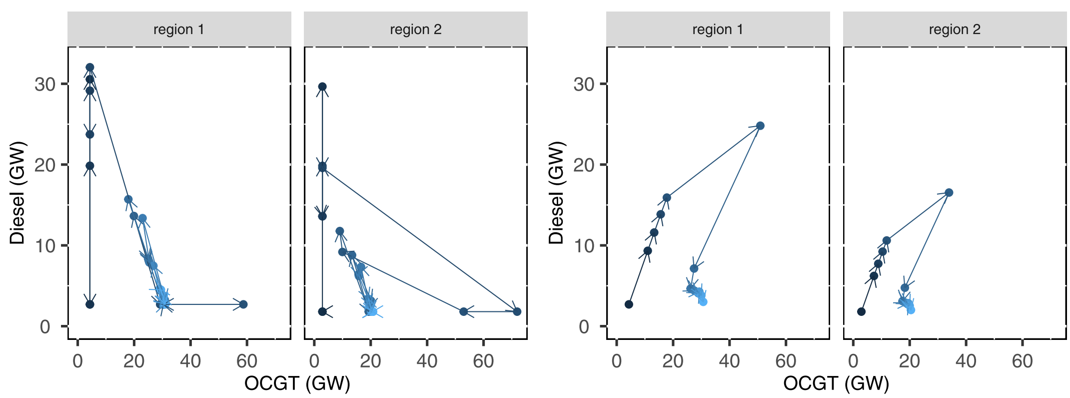
$$p^G_{gt}, p^{AccG}_g, p^{ShedP}_{zt}, p^{SE+}_{st}, p^{SE-}_{st}, p^{AccSE}_s, q^{SE}_{st}, p^{AccR}_r, p^{GShedP}_{zt} \in \mathbb{R}^+_0. \tag{6l}$$

The operational cost function $c^{OPE}(\mathbf{x}, \mathbf{c})$ includes the total operating costs of all generators, $C^G_g p^G_{gt}$, and storage facilities $C^{SE}_s p^{SE+}_{st}$ and load shedding costs $C^{ShedP} p^{ShedP}_{zt}$. The parameters C^G_g and C^{SE}_s include the variable operational cost of generators and storage. For thermal generators, C^G_g also includes the fuel cost and the CO_2 tax charged on the emissions of generators. Constraint (6b) ensures that generators are within their capacity limits. Constraint (6c) ensures that the power flow p^L_{lt} is within the line transmission capacity p^{AccL}_l. Constraints (6d) and (6e) dictate that the charging power p^{SE+}_{st} and the discharging power p^{SE-}_{st} of a storage facility should be within the capacity, respectively. Constraint (6f) limits the energy storage level q^{SE}_{st} to be within the capacity q^{AccSE}_s. Constraint (6g) captures how fast thermal generators can ramp up or ramp down their power output. The parameters α^G_g is the maximum ramp rate of thermal generators. The power nodal balance, Constraint (6h), ensures that in one operational period t, the sum of total power generation of thermal generators p^G_{gt}, power discharged from all the electricity storage p^{SE-}_{st}, renewable generation $R^R_{zt} p^{AccR}_{rt}$, power transmitted to this region, and load shed p^{ShedP}_{zt} equals the sum of power demand $\mu^{DP} P^{DP}_{zt}$ power transmitted to other regions, and power generation shed p^{GShedP}_{zt}. The parameter R^R_{rt} is the capacity factor of a renewable unit that is a fraction of the nameplate capacity p^{AccR}. Constraint (6i) states that the state of charge q^{SE}_{st} in period $t+1$ depends on the previous state of charge q^{SE}_{st}, the charged power p^{SE+}_{st} and discharged power p^{SE-}_{st}. The parameter η^{SE}_s represent the charging efficiency. Constraint (6j) restricts the total emission: the parameter H_t is the length of the period t, and E^G_g is the emission per unit of power generated. The capacities p^{AccG}_g, p^{AccL}_l, p^{AccSE}_s, scaling factor of demand μ^{DP} and CO_2 budget μ^E are passed from the master problem, Eqs. (5), via vector $\mathbf{x}_i$ and CO_2 tax that is included in cost coefficient C^G_g is passed from master problem, Eqs. (5), via vector $\mathbf{c}_i$.



(a) Case A (unstabilised Benders, 13 iters) (b) Case A (stabilised Benders, 13 iters)

Fig. 5. Comparative results of Case A.



(a) Case B (unstabilised Benders, 26 iters) (b) Case B (stabilised Benders, 13 iters)

Fig. 6. Comparative results of Case B.

Table 1
Summary of the illustrative cases.

	Description
Case A	Single region, two technologies to invest (OCGT and Diesel)
Case B	Two unconnected regions with sizes 60% and 40% of case A
Case C	Case B with a transmission line with 0 initial capacity

6. Results

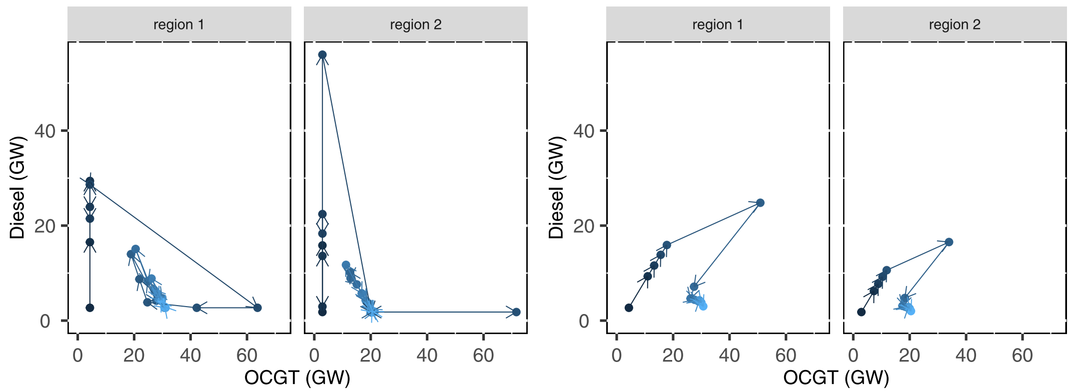
This section first uses small illustrative cases to show how stabilisation helps solve multi-region investment planning problems. Then we demonstrate the proposed algorithm on larger instances and present the computational results.

6.1. Illustrative cases

We use three cases to show the value of stabilisation in a multi-region investment planning problem. A summary of the four cases is presented in Table 1. To simplify the visualisation of the results, we consider only two types of generation, OCGT and Diesel, and a one-time investment planning problem is solved. This means that there is no long-term uncertainty, and there is only one short-term operational problem. In this case, there is no difference between AB and standard Benders because there is only one **SP**.

Figs. 5–7 show how solutions are explored until convergence. In each figure, the darkest blue point represents the initial solution, the lightest blue point is the optimal solution, and the arrows indicate the order of points explored. For the stabilised versions, the stabilisation factor, γ, is fixed to 0.2. In all cases, there is degeneracy in the dimension of the total amounts of the two generation types. From Case B and Case C, we find that there is degeneracy in the dimension of regions. In the two region cases, there is a CO_2 constraint that restricts the total emissions from both regions. In Case B, where to put the capacities becomes relevant. From Fig. 6, we see that without stabilisation, the algorithm struggles to balance the capacities of the two technologies and starts jumping to points with different proportions of the two technologies many times until it finds the optimal solution. In Fig. 6, we see that the stabilised approach is clearer about which direction to explore and make small movements towards the optimal instead of sampling points wildly. The number of iterations is doubled without stabilisation. For a more realistic problem with more technologies and regions and more complicated network topology, the value of stabilisation reveals further, as is shown in Section 6.3.

In Case C, two regions are initially disconnected, but a line can be invested to connect them. However, there should be no line invested because the two regions are proportional to each other and making investments in the local generation is optimal. By observing the solution proposed by **RMP** in the unstabilised version, we notice that **RMP** does not realise that and invests in the line in some iterations before finding the optimum, and this leads to more iterations compared with the stabilised version.



(a) Case C (unstabilised Benders, 27 iters) (b) Case C (stabilised Benders, 13 iters)

Fig. 7. Comparative results of Case C.

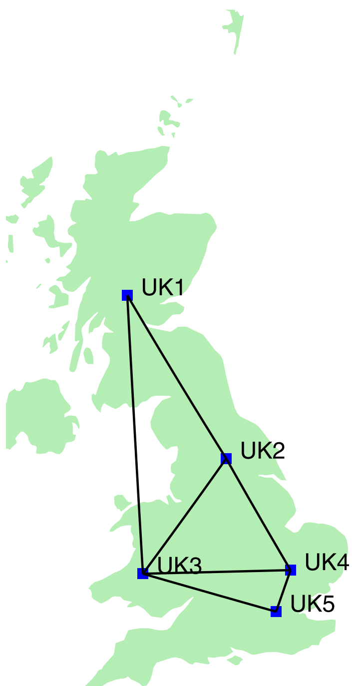


Fig. 8. Illustration of the UK power system. (UK1: Scotland, UK2: North England, UK3: Midland and Wales, UK4: East England and UK5: South England.)

6.2. Case study

We test the SAB algorithm with adaptive oracles on the stochastic investment planning of the UK power system. We use the model presented in Section 5 to investigate the computational issues. The network topology is shown in Fig. 8. We implemented the algorithm and model in Julia 1.7.3 using JuMP (Dunning et al., 2017) and solved with Gurobi 9.5.1 (Gurobi Optimization, LLC, 2022). We ran the code on nodes of a computer cluster with a 2x 3.6 GHz 8 core Intel Xeon Gold 6244 CPU and 384 GB of RAM, running on CentOS Linux 7.9.2009. Some data was taken from Mazzi et al. (2020).

6.3. Computational results

This section presents the computational results of the proposed SAB. We first compare the performance of SAB against the unstabilised versions of AB with one **SP** solved per iteration (Mazzi et al., 2020) and the standard Benders. We use the model presented in Section 5 to solve a 5-region UK power system planning to make the benchmark. The long-term uncertainties are CO_2 price, CO_2 budget and power demand. The short-term uncertainties are the wind and solar capacity factors and load profiles. The summary of cases and their problem sizes are shown in Table 2. In all case examples Case 0–3, there are four short-term operational scenarios, each consisting of 4380 operational periods. Case 0 has no long-term uncertainty. Case 1 has one long-term uncertainty, CO_2 budget. Case 2 has CO_2 budget and long-term demand uncertainty. Case 3 has CO_2 budget, long-term demand, and CO_2 tax as long-term uncertainty.

From Table 3, we can see that (a) SAB is up to 113.7 times faster than standard Benders for a 1.00% convergence tolerance and 45.5 times faster than standard Benders for a 0.10% convergence tolerance, (b) AB gets slower when converging to a tighter tolerance and (c) compared with AB, SAB is up to 2.1 times faster for a 1.00% convergence tolerance, and AB cannot solve the largest instance to 0.10% due to severe oscillation. Therefore, for Case 3, we report the performance of unstabilised AB when it reaches a tolerance of 0.103%, which is the tightest convergence tolerance it achieves just before it starts oscillating severely. Gurobi can only solve Case 0, taking 440 s, but cannot solve the other cases. The advantage of SAB over AB is due to two factors: the level method stabilisation and the adaptive selection of the number of subproblems to solve each iteration. Table 3 also shows that the results for AB with multiple subproblem solves. From this, we see that both the multiple subproblem solves and the stabilisation contribute significantly to the improved performance of SAB compared to AB.

For Cases 0–3, we measure the total distance the RMP point moved in AB, and the total distance the LMP point moved in SAB. The total distance is a metric to evaluate the oscillation. By comparing the distances from AB and SAB, we notice that the total distance LMP moved is much less than the total distance RMP point moved in AB, especially when the algorithms converge from 1% to 0.1% convergence tolerance. This means that SAB reduces the oscillation, which aligns with the illustrative examples in Section 6.1.

6.3.1. Improving the robustness

The stabilisation factor γ significantly impacts the performance. A very small γ leads to loose stabilisation and makes stabilisation less effective, whereas a very large γ leads to tight stabilisation and may hinder the exploitation of the solution space. We test the performance using different γ from 0.025 to 0.9 and present the results in Table 4, and we find that $\gamma = 0.025$ gives the best performance on average. However, we find that a stabilisation factor of less than 0.2 is significantly better and generally performs well. For different cases, the γ that yields better performance varies. A common rule of thumb for setting a fixed stabilisation factor may be to set it to 0.5 (Zverovich et al., 2012).

Table 2
Overview of the cases used in the computational study.

	Operational periods per short-term scenario	Short-term scenarios	Long-term scenarios	Number of decision nodes				Problem size (undecomposed)		
				Present	In 5 years	In 10 years	Total	Variables	Constraints	Nonzeros
Case 0	4380	4	1	1	1	1	3	2.7×10^6	7.7×10^6	1.9×10^7
Case 1	4380	4	9	1	3	9	13	1.6×10^7	4.6×10^7	1.1×10^8
Case 2	4380	4	81	1	9	81	91	1.2×10^8	3.5×10^8	8.4×10^8
Case 3	4380	4	729	1	27	729	757	1.0×10^9	4.5×10^9	1.1×10^{10} [a]

[a] The model cannot be loaded into the solver due to its size.

Table 3
Comparative results for standard Benders, AB and SAB, γ is fixed to 0.025 for SAB. (speed up: the time spent using standard Benders divided by the time spent using AB or SAB).

	ϵ (%)	Standard benders		AB			AB with adaptive subproblem selection			SAB		
		Iters/Evals	Time (s)	Iters/Evals	Time (s)	Speed up	Iters/Evals	Time (s)	Speed up	Iters/Evals	Time (s)	Speed up
Case 0	1.00	18/36	1 051	30/31	874	1.20	26/35	970	1.08	16/26	751	1.40
	0.10	33/66	1 925	66/67	1 925	1.00	51/61	1 754	1.10	35/47	1 344	1.43
Case 1	1.00	16/192	5 698	32/33	953	5.98	21/45	1 308	4.36	20/27	791	7.20
	0.10	28/336	9 922	61/62	1 823	5.44	37/46	1 343	7.39	28/40	1 156	8.58
Case 2	1.00	11/990	30 532	54/55	1 559	19.58	26/52	1 396	21.87	23/60	1 662	18.37
	0.10	18/1620	48 666	173/174	4 982	9.77	45/128	3 725	13.06	41/106	3 100	15.70
Case 3	1.00	16/12 096	382 828	202/203	7 203	53.10	25/188	5 341	71.68	25/188	3 367	113.70
	0.10	3736/18 144	563 205	3736/3737 [a]	422 591 [a]	<1.33 [a]	56/626	21 070	>26.73	72/360	12 375	>45.51

[a] The algorithm cannot solve the problem to a 0.1% tolerance but reach a 0.103% tolerance.

Table 4
Results of SAB decomposition with different level sets.

γ	ϵ (%)	Case 0		Case 1		Case 2		Case 3		Average	
		Iters/Evals	Time (s)	Iters/Evals	Time (s)	Iters/Evals	Time (s)	Iters/Evals	Time (s)	Iters/Evals	Time (s)
0.000	1.00	35/39	1106	21/45	1298	23/70	1 925	24/169	7 270	26/81	11 599
0.025	1.00	16/26	752	20/27	791	23/60	1 662	30/105	3 385	22/55	6 590
0.050	1.00	13/20	552	20/28	813	34/74	2 039	21/163	4 511	22/71	7 915
0.075	1.00	16/23	665	15/21	581	21/64	1 859	30/232	10 381	21/85	13 486
0.100	1.00	12/20	615	34/62	1820	17/60	1 694	31/164	5 419	24/77	9 548
0.200	1.00	18/24	657	35/64	1885	31/90	2 588	46/180	12 339	33/90	17 469
0.300	1.00	23/30	848	30/50	1478	33/88	2 503	68/153	5 055	39/80	9 884
0.400	1.00	22/29	842	37/58	1796	39/96	2 755	140/633	22 585	60/204	27 978
0.500	1.00	24/32	939	31/38	1276	58/143	4 180	358/833	91 031	118/262	97 426
0.600	1.00	24/31	980	44/57	1721	67/110	3 457	566/1123	45 590	175/330	51 748
0.700	1.00	32/37	1146	75/90	2970	126/212	6 665	551/977	40 753	196/329	51 534
0.800	1.00	39/45	1424	84/94	3438	197/275	9 004	713/966	43 529	258/345	57 395
0.900	1.00	73/81	2739	222/244	8165	487/649	21 665	2149/2859	192 864	733/958	225 433
0.000	0.10	55/59	1696	37/86	2508	41/116	3 354	59/473	34 296	48/184	41 854
0.025	0.10	35/47	1344	28/40	1156	41/106	3 100	72/360	12 472	44/138	18 072
0.050	0.10	21/33	941	29/43	1252	52/124	3 693	62/502	16 862	41/176	22 748
0.075	0.10	24/35	1022	27/37	1051	35/109	3 181	81/690	49 078	42/218	54 332
0.100	0.10	21/33	996	44/81	2384	33/105	3 037	60/400	16 605	40/155	23 022
0.200	0.10	26/36	1004	54/97	2880	54/153	4 556	92/510	40 627	57/199	49 067
0.300	0.10	31/43	1227	44/78	2322	61/88	7 101	140/559	19 737	69/192	30 387
0.400	0.10	27/37	1074	49/78	2395	53/174	5 279	306/1499	59 641	109/447	68 389
0.500	0.10	31/42	1234	39/50	1644	77/239	7 181	471/1217	108 217	155/387	118 276
0.600	0.10	31/41	1286	60/84	2513	112/211	6 674	838/2054	93 282	260/598	103 755
0.700	0.10	45/51	1555	89/115	3717	178/212	9 965	954/2256	109 567	317/659	124 804
0.800	0.10	50/62	1923	104/130	4523	326/599	18 881	1085/1848	94 686	391/660	120 013
0.900	0.10	85/94	3160	233/262	8709	657/1015	33 098	2982/4706	376 947	989/1519	421 914

We see that different fixed γ can lead to a noticeable difference in performance, which aligns with Remark 5 in Zverovich et al. (2012). This gives the motivation to test the approach presented in Section 3.1 of adjusting γ with the aim of making the level method stabilisation more robust and independent of the choice of the stabilisation factor.

We test extensively the dynamic stabilisation scheme on Case 0–3. The results for different cases are shown in Table 5. We can see that dynamic stabilisation can sometimes outperform the best performance with fixed stabilisation. By comparing results from Table 5 and results using fixed stabilisation factors in Table 4, we can see that the dynamic stabilisation makes the level method stabilisation much more robust in terms of the choice of γ. It is particularly valuable because one may need extensive tests to find the γ that yields the best performance for the problems to be solved. However, as we see in Table 4, different problems may have different best γ. Therefore, a dynamic stabilisation that makes the performance less dependent on the choice of γ may make it easier to get a satisfying performance if one chooses a bad initial γ because the dynamic adjustment will help correct γ to a sensible value while solving the problem.

6.4. Power system analysis

In this section, we present the results of the 5-region UK power system planning problem. We analyse the investment decisions, expected costs, and the Value of the Stochastic Solution (VSS).

The investment decisions in the first stage are presented in Table 6. There are no investments in technologies except the onshore wind in the first investment stage. We notice that the transmission lines

Table 5
Results for stabilised Benders decomposition with adjusted level sets (speed up: the time spent using fixed $\gamma = 0.025$ divided by time spent using dynamic stabilisation).

	Initial γ	ω	$\underline{P}$	$\overline{P}$	Iters/Evals		Time (s)			
					$\epsilon = 1.00\%$	$\epsilon = 0.10\%$	$\epsilon = 1.00\%$	Speed up	$\epsilon = 0.10\%$	Speed up
Case 0	0.025	0.5	0.1	0.9	20/30	35/48	843	0.89	1 368	0.98
		0.9	0.1	0.9	17/27	28/44	702	1.07	1 233	1.09
	0.1	0.5	0.1	0.9	14/19	31/38	526	1.43	1 081	1.24
		0.9	0.1	0.9	13/18	23/31	484	1.55	862	1.56
	0.5	0.5	0.1	0.9	17/24	27/37	753	1.00	1 212	1.11
		0.9	0.1	0.9	20/27	28/38	761	0.99	1 112	1.21
	0.9	0.5	0.1	0.9	31/39	53/43	1123	0.67	1 554	0.86
		0.9	0.1	0.9	33/41	43/55	1248	0.60	1 655	0.81
Case 1	0.025	0.5	0.1	0.9	22/31	35/49	882	0.90	1 425	0.81
		0.9	0.1	0.9	17/27	28/43	732	1.08	1 238	0.93
	0.1	0.5	0.1	0.9	14/27	31/57	753	1.05	1 651	0.70
		0.9	0.1	0.9	18/28	26/40	734	1.08	1 157	1.00
	0.5	0.5	0.1	0.9	22/42	31/54	1182	0.67	1 599	0.72
		0.9	0.1	0.9	22/45	34/45	1254	0.63	2 054	0.56
	0.9	0.5	0.1	0.9	15/20	30/44	627	1.26	1 342	0.86
		0.9	0.1	0.9	26/41	35/61	1277	0.62	1 901	0.61
Case 2	0.025	0.5	0.1	0.9	38/68	51/110	1979	0.84	3 224	0.96
		0.9	0.1	0.9	18/61	41/152	1601	1.04	4 461	0.69
	0.1	0.5	0.1	0.9	28/67	36/94	1775	0.94	2 720	1.14
		0.9	0.1	0.9	29/67	39/99	1931	0.86	2 858	1.08
	0.5	0.5	0.1	0.9	39/90	46/108	2533	0.66	3 191	0.97
		0.9	0.1	0.9	39/102	49/141	3124	0.53	4 261	0.73
	0.9	0.5	0.1	0.9	48/95	76/194	2703	0.61	5 839	0.53
		0.9	0.1	0.9	50/107	65/164	3377	0.49	5 086	0.61
Case 3	0.025	0.5	0.1	0.9	49/217	84/418	6573	0.51	14 067	0.88
		0.9	0.1	0.9	28/173	66/399	5486	0.61	15 195	0.81
	0.1	0.5	0.1	0.9	24/111	87/433	3306	1.02	14 860	0.83
		0.9	0.1	0.9	44/143	126/606	4538	0.74	21 276	0.58
	0.5	0.5	0.1	0.9	52/182	140/566	5837	0.58	24 892	0.50
		0.9	0.1	0.9	61/210	123/494	6901	0.49	17 332	0.71
	0.9	0.5	0.1	0.9	58/194	98/425	5758	0.58	14 838	0.83
		0.9	0.1	0.9	76/178	133/444	6042	0.56	15 820	0.78
Average								0.83		0.87

Table 6
Investment in onshore wind in the first investment stage.

	Investment in onshore wind (GW)					
	Scotland	North England	Midlands & Wales	East England	South England	UK total
Case 0	0.00	25.70	0.00	0.00	65.15	90.85
Case 1	14.76	28.25	0.00	0.00	51.68	94.69
Case 2	3.56	23.22	0.00	0.00	59.19	85.97
Case 3	1.42	23.60	0.00	0.00	69.16	94.18

are expanded in the later investment nodes. Therefore, for the first investment stage, only investment in onshore wind is presented. The onshore wind is mainly invested in Scotland, North England and South England in the first investment stage. When considering only short-term uncertainty, we can see that in Case 0, a total of 90.85 GW of onshore wind is invested, 28% of which is in North England. Compared with Case 1, around 3.8 GW less capacity is installed in Case 0. When considering uncertainty in both long-term demand and CO_2 budget, we can see a 3.56 GW investment in onshore wind in Scotland, compared with 14.76 GW in Case 1 and 1.42 GW in Case 3.

Table 7 shows the optimal costs and the VSS for considering long-term uncertainties. We can see that there is up to £7702 million VSS when considering uncertainty, including CO_2 budget and long-term demand. The VSS is £2904 million when considering only CO_2 budget as an uncertainty parameter. When considering long-term uncertainty, including CO_2 budget, CO_2 tax and long-term demand, the VSS is 4.4% of the optimal cost. This shows the value of including long-term uncertainty in a long-term planning problem and solving a large model.

Table 7
Optimal costs and VSS.

	Case 0	Case 1	Case 2	Case 3
Optimal cost (mn £)	174 099	174 276	174 871	174 785
VSS (long-term uncertainty, mn £)	–	2 904	7 325	7 702

7. Discussion

In this paper, we propose a Benders-type decomposition method to address the computational difficulty of multi-horizon stochastic programming with short-term and long-term uncertainty by exploiting its unique structure. Compared with other solution methods (Zakeri et al., 2000; Downward et al., 2020), we exploit the properties of the **SP** and stabilise the algorithm with the level method and adaptively select **SP**s to solve exactly per iteration for a better approximation, which shows significant performance improvement. The method can be generally applied to solve any problem that is formulated as Eqs. (1)–(2).

We demonstrate our proposed method on a multi-region UK power system planning problem. To the authors' knowledge, this is the first study that presents a multi-horizon formulation of a multi-region power system planning problem with short-term and long-term uncertainty and proposes a method to solve such a problem efficiently. Compared with a similar problem for long-term investment planning such as Backe et al. (2022) that only considers short-term uncertainty, this paper firstly introduces both long-term and short-term uncertainty in a power system planning problem using a multi-horizon framework.

We notice and analyse the oscillation of the Benders-type decomposition method for multi-region investment planning problems. The level method stabilisation approach was used to stabilise Benders. Compared with the existing literature that studied the level method, we integrate it with the inexact oracles and show that it significantly improves computational performance. In addition, similar studies usually set the level set in an ad hoc way (Zverovich et al., 2012; Ruszczyński and Świętanowski, 1997). Moreover, we test to adjust the target based on a proposed measurement. For the test instance, adjusting the level set can usually yield better or equivalent performance. However, the parameters that yield the best performance may be case-dependent.

Although the stabilisation is helpful, the stabilisation problem can potentially be a large QP and be slow to solve. One possible approach to stabilise the problem efficiently is to utilise the built-in method, analytic centre (Gondzio et al., 1996) in a commercial solver like Gurobi to solve a feasibility problem to avoid solving a QP. We test utilising the analytic centre of Gurobi to avoid solving a QP **LMP**. However, the results show that proper stabilisation may still be the better option, even for large problems. Another approach is to use the L1 or L-infinity norm and linearise the problem. The alternative norms and linearisation have been tested, but the performance is not as good as using the L2 norm. We observed that the slower performance using other norms is because it leads to significantly more iterations compared with using the L2 norm, although the **LMP** is linear programming and cheaper to solve. It may be worth investigating which norm is suitable for different problems and the reasoning behind it in the future.

We demonstrated the method for solving large-scale linear programming. However, the method can be applied to solve mixed-integer linear programming problems without modification as long as the integer variables are in the **RMP**. In such a case, the stabilisation problem becomes a mixed-integer QP problem which may be slow to solve. Some other stabilisation techniques, such as local branching (Baena et al., 2020) may be an alternative.

Although this paper presents a general method to solve a class of large-scale optimisation problems very efficiently, a limitation is that we need the same coefficient matrices in all nodes to utilise the adaptive oracles. This may be limited when different operational scenarios are preferred. However, the adaptive oracles can be used within each group of nodes with the same matrices and with different oracles for each different group.

8. Conclusions and future work

In this paper, we proposed stabilised Benders decomposition with adaptive oracles to efficiently solve a class of large-scale linear programming problems and address the degeneracy issue. We apply the algorithm to solve a multi-stage stochastic programming problem with short-term and long-term uncertainty. The stochastic programming problem is a multi-region UK power system investment planning problem towards 2035. The multi-region part of the problem leads to high degeneracy and oscillation. The test instances have up to 1 billion variables and 4.5 billion constraints. The computational results show that: (a) for a 1.00% convergence tolerance, the proposed stabilised method is up to 113.7 times faster than standard Benders decomposition and 2.1 times faster than AB decomposition without stabilisation; (b) for a 0.10% convergence tolerance, the proposed stabilised method is up to 45.5 times faster than standard Benders decomposition and the unstabilised AB decomposition cannot solve the largest instance to the convergence tolerance due to severe oscillation; (c) dynamic level method increases the robustness of the stabilisation. We note that both the adaptive selection of subproblems and the level method stabilisation contribute significantly to the improvement.

Although the proposed method reduced the computational effort significantly and was used to solve multi-horizon stochastic programming with short-term and long-term uncertainty, we notice that for a very large problem with many decision nodes, the reduced master problem and the stabilisation problem may take longer to solve. Therefore, in future, techniques including node aggregation and cuts selection and deletion and stronger cuts generation (Oliveira et al., 2014) may be needed to improve the performance. It is also possible to combine Lagrangean decomposition (Escudero et al., 2016) with Benders decomposition when solving huge problems. In addition, although multi-horizon formulation reduces the problem size significantly, the model size may be reduced further by adjusting the scenario tree, e.g., removing the scenarios that do not make a difference while solving the problem.

CRediT authorship contribution statement

Hongyu Zhang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Nicolò Mazzi:** Software, Methodology, Conceptualization. **Ken McKinnon:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Rodrigo Garcia Nava:** Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Asgeir Tomasgard:** Writing – review & editing, Writing – original draft, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the Research Council of Norway through PETROSENTER LowEmission [project code 296207].

References

Backe, S., Skar, C., del Granado, P.C., Turgut, O., Tomasgard, A., 2022. EMPIRE: An open-source model based on multi-horizon programming for energy transition analyses. SoftwareX 17, 100877. http://dx.doi.org/10.1016/j.softx.2021.100877.

Baena, D., Castro, J., Frangioni, A., 2020. Stabilized benders methods for large-scale combinatorial optimization, with application to data privacy. Manage. Sci. 66, 3051–3068. http://dx.doi.org/10.1287/mnsc.2019.3341.

Benders, J.F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numer. Math. 4, 238–252. http://dx.doi.org/10.1007/BF01386316.

Birge, J.R., 1985. Decomposition and partitioning methods for multistage stochastic linear programs. Oper. Res. 33, 989–1007. http://dx.doi.org/10.1287/OPRE.33.5.989.

Birge, J.R., Louveaux, F., 2011. Introduction to stochastic programming. Springer Science & Business Media, http://dx.doi.org/10.1007/978-1-4614-0237-4.

Boffino, L., Conejo, A.J., Sioshansi, R., Oggioni, G., 2019. A two-stage stochastic optimization planning framework to decarbonize deeply electric power systems. Energy Econ. 84, 104457. http://dx.doi.org/10.1016/J.ENECO.2019.07.017.

Downward, A., Baucke, R., Philpott, A.B., 2020. JuDGE.jl: A Julia Package for Optimizing Capacity Expansion. Technical Report Engineering Science, University of Auckland, pp. 1–26, URL http://www.optimization-online.org/DB_FILE/2020/11/8086.pdf.

Dunning, I., Huchette, J., Lubin, M., 2017. JuMP: A modeling language for mathematical optimization. SIAM Rev. 59, 295–320. http://dx.doi.org/10.1137/15M1020575.

Escudero, L.F., Garín, M.A., Unzueta, A., 2016. Cluster Lagrangean decomposition in multistage stochastic optimization. Comput. Oper. Res. 67, 48–62. http://dx.doi.org/10.1016/J.COR.2015.09.005.

Fabian, C.I., 2000. Bundle-type methods for inexact data. CEJOR Cent. Eur. J. Oper. Res. 8, 35–55, URL https://www.researchgate.net/profile/Csaba-Fabian/publication/228468981_Bundle-type_methods_for_inexact_data/links/0f3175374f931038c8000000/Bundle-type-methods-for-inexact-data.pdf.

Fairbrother, J., Turner, A., Wallace, S.W., 2022. Problem-driven scenario generation: An analytical approach for stochastic programs with tail risk measure. Math. Program. 191, 141–182. http://dx.doi.org/10.1007/s10107-019-01451-7.

Fletcher, R., 2000. Practical Methods of Optimization. John Wiley & Sons, Ltd, http://dx.doi.org/10.1002/9781118723203.

Fodstad, M., Egging, R., Midthun, K., Tomasgard, A., 2016. Stochastic modeling of natural gas infrastructure development in europe under demand uncertainty. Energy J. 37, 5–32. http://dx.doi.org/10.5547/01956574.37.SI3.mfod.

Gacitua, L., Gallegos, P., Henriquez-Auba, R., Lorca, Negrete-Pincetic, M., Olivares, D., Valenzuela, A., Wenzel, G., 2018. A comprehensive review on expansion planning: Models and tools for energy policy analysis. Renew. Sustain. Energy Rev. 98, 346–360. http://dx.doi.org/10.1016/j.rser.2018.08.043.

Galan, A., de Prada, C., Gutierrez, G., Sarabia, D., Grossmann, I.E., Gonzalez, R., 2019. Implementation of RTO in a large hydrogen network considering uncertainty. Optim. Eng. 20, 1161–1190. http://dx.doi.org/10.1007/s11081-019-09444-3.

Gondzio, J., du Merle, O., Sarkissian, R., Vial, J.P., 1996. ACCPM - A library for convex optimization based on an analytic center cutting plane method. European J. Oper. Res. 94, 206–211. http://dx.doi.org/10.1016/0377-2217(96)00169-5.

Gupta, V., Grossmann, I.E., 2014. Multistage stochastic programming approach for offshore oilfield infrastructure planning under production sharing agreements and endogenous uncertainties. J. Pet. Sci. Eng. 124, 180–197. http://dx.doi.org/10.1016/j.petrol.2014.10.006.

Gurobi Optimization, LLC, 2022. Gurobi optimizer reference manual. URL https://www.gurobi.com.

Hellemo, L., Midthun, K., Tomasgard, A., Werner, A., 2013. Multi-stage stochastic programming for natural gas infrastructure design with a production perspective. In: Stochastic Programming: Applications in Finance, Energy, Planning and Logistics. World Scientific, pp. 259–288. http://dx.doi.org/10.1142/9789814407519{_}0010.

Kaut, M., Midthun, K.T., Werner, A.S., Tomasgard, A., Hellemo, L., Fodstad, M., 2014. Multi-horizon stochastic programming. Comput. Manag. Sci. 11, 179–193. http://dx.doi.org/10.1007/s10287-013-0182-6.

King, A., Wallace, S., 2012. Modeling with Stochastic Programming. Springer Science & Business Media, http://dx.doi.org/10.1007/978-0-387-87817-1.

Lara, C.L., Siirola, J.D., Grossmann, I.E., 2020. Electric power infrastructure planning under uncertainty: Stochastic dual dynamic integer programming (SDDiP) and parallelization scheme. Optim. Eng. 21, 1243–1281. http://dx.doi.org/10.1007/s11081-019-09471-0.

Lemarechal, C., Nemirovskii, A., Nesterov, Y., 1995. New variants of bundle methods. Math. Program. Ser. B 69, 111–147. http://dx.doi.org/10.1007/BF01585555.

Li, C., 2021. Algorithms for Stochastic Mixed-Integer Nonlinear Programming and Long Term Optimization of Electric Power Systems (Ph.D. thesis). Carnegie Mellon University, http://dx.doi.org/10.1184/R1/19146431.V1.

Li, C., Conejo, A.J., Liu, P., Omell, B.P., Siirola, J.D., Grossmann, I.E., 2022. Mixed-integer linear programming models and algorithms for generation and transmission expansion planning of power systems. European J. Oper. Res. 297, 1071–1082. http://dx.doi.org/10.1016/j.ejor.2021.06.024.

Mazzi, N., Grothey, A., McKinnon, K., Sugishita, N., 2020. Benders decomposition with adaptive oracles for large scale optimization. Math. Program. Comput. http://dx.doi.org/10.1007/s12532-020-00197-0.

Munoz, F.D., Hobbs, B.F., Watson, J.P., 2016. New bounding and decomposition approaches for MILP investment problems: Multi-area transmission and generation planning under policy constraints. European J. Oper. Res. 248, 888–898. http://dx.doi.org/10.1016/j.ejor.2015.07.057.

Munoz, F.D., Watson, J.P., 2015. A scalable solution framework for stochastic transmission and generation planning problems. Comput. Manag. Sci. 12, 491–518. http://dx.doi.org/10.1007/s10287-015-0229-y.

Oliveira, F., Grossmann, I.E., Hamacher, S., 2014. Accelerating benders stochastic decomposition for the optimization under uncertainty of the petroleum product supply chain. Comput. Oper. Res. 49, 47–58. http://dx.doi.org/10.1016/J.COR.2014.03.021.

Pereira, M.V., Pinto, L.M., 1991. Multi-stage stochastic optimization applied to energy planning. Math. Program. 52, 359–375. http://dx.doi.org/10.1007/BF01582895/METRICS.

Powell, W.B., Meisel, S., 2016. Tutorial on stochastic optimization in energy - Part I: Modeling and policies. IEEE Trans. Power Syst. 31, 1459–1467. http://dx.doi.org/10.1109/TPWRS.2015.2424974.

Rahmaniani, R., Crainic, T.G., Gendreau, M., Rei, W., 2017. The Benders decomposition algorithm: A literature review. European J. Oper. Res. 259, 801–817. http://dx.doi.org/10.1016/j.ejor.2016.12.005.

Ruszczyński, A., Świętanowski, A., 1997. Accelerating the regularized decomposition method for two stage stochastic linear problems. European J. Oper. Res. 101, 328–342. http://dx.doi.org/10.1016/S0377-2217(96)00401-8.

Salo, A., Andelmin, J., Oliveira, F., 2022. Decision programming for mixed-integer multi-stage optimization under uncertainty. European J. Oper. Res. 299, 550–565. http://dx.doi.org/10.1016/J.EJOR.2021.12.013.

Singh, K.J., Philpott, A.B., Kevin Wood, R., 2009. Dantzig-wolfe decomposition for solving multistage stochastic capacity-planning problems. Oper. Res. 57, 1271–1286. http://dx.doi.org/10.1287/opre.1080.0678.

Skar, C., Doorman, G., Tomasgard, A., 2014. Large-scale power system planning using enhanced Benders decomposition. In: Proceedings - 2014 Power Systems Computation Conference, PSCC 2014. Power Systems Computation Conference, http://dx.doi.org/10.1109/PSCC.2014.7038297.

Slyke, R.M.V., Wets, R., 1969. Optimal control and stochastic programming. SIAM J. Appl. Math. 17, 638–663. http://dx.doi.org/10.1137/0117061.

Wallace, S.W., Fleten, S.E., 2003. Stochastic programming models in energy. Handbooks Oper. Res. Management Sci. 10, 637–677. http://dx.doi.org/10.1016/S0927-0507(03)10010-2.

Zakeri, G., Philpott, A., Ryan, D., 2000. Inexact cuts in Benders decomposition. SIAM J. Optim. 10, 643–657. http://dx.doi.org/10.1137/S1052623497318700.

Zverovich, V., Fábián, C.I., Ellison, E.F., Mitra, G., 2012. A computational study of a solver system for processing two-stage stochastic LPs with enhanced Benders decomposition. Math. Program. Comput. 4, 211–238. http://dx.doi.org/10.1007/s12532-012-0038-z.